UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/748,427	12/30/2003	Dan M. White	Intel-017PUS	8110

7590 09/12/2007

Daly, Crowley & Mofford, LLP
c/o PortfolioIP
P.O. Box 52050
Minneapolis, MN 55402

EXAMINER	
WU, JUNCHUN	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
09/12/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/748,427	WHITE, DAN M.	
	Examiner	Art Unit	
	Junchun Wu	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 June 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3,5,7,9-13,15-18,20 and 24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,5,7,9-13,15-18,20 and 24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the amendment filed on June 21, 2007.

2. Claims 4, 6, 8, 14, 19 and 21-23 have been cancelled.

3. Claims 1-3, 5, 9, 12, 13, 15, 16, 18, 20, and 24 have been amended.

4. Claims 1-3, 5, 7, 9-13, 15-18, 20 and 24 are pending.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-3, 5, 7, and 12,13, and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bade et al. (US Pub. No.20020059054 A1, hereinafter "Bade"), in view of Mulchandani et al. (US Patent No. 5,701,488, hereinafter "Mulchandani").

7. Per claim 1 (Currently Amended)

Bade discloses

- A method of displaying embedded firmware program information ([0021] "An integrated design environment (IDE) is disclosed for simulating embedded systems")

Comprising:

- displaying a first screen to interact with a user for high level function selections ([0101] *"As shown in **FIG**.21the IDE preferably has a menu-driven graphical user interface that*

preferably includes a design window for creating a design with toolbars for accessing functions using a computer mouse or similar interface. The IDE preferably includes a peripheral design editor and simulator that is adapted to permit hardware IP components and processes to be created and linked with other IP components.").

- displaying a second screen to show hardware resources for a programmable circuit ([0099] *"As shown in **FIGS. 28, 29, and 37**, the use of an instruction set accurate simulator to model a processor core permits the processor simulator to exchange memory transactions with the hardware partition and to receive interrupts from the hardware partition using APIs linking the hardware partition and the instruction set accurate simulator."*).
- displaying a third screen to show source code for a plurality of source code programs to control the programmable circuit ([0173] *"The IDE preferably features a high-quality C++ code generator, hiding all the details of generating simulation code..."*)
- displaying a fourth screen to render symbolic information associated with the displayed source code ([0104] *"**FIG. 36B** shows an example of a software debugger interface window **3685** superimposed over a design window **3620** of a virtual embedded system." & In Fig. 36B shows the design window 3620 associated with source code that is debugging in debugger window 3620*).

The symbolic information comprising:

- listings including named registers, data labels for word, byte and short entities ([0163] *"As shown in **FIG. 26, a** Test Bench Builder Toolbar is preferably included to represent the test bench controls, such as a LED, LCD, memory viewer, ASCII terminal window,*

resource meter, or signal button, and the test bench builder, allows a user to quickly add these controls to a test bench." ; the *memory viewer* may view data words, short words and byte values; and *register viewer* may list information associated with registers).

But Bade does not disclose

- code labels, data labels referring to data structures, data register names, and index register names; address locations for the code labels and the data labels; and names of the data structures individually expandable to show addresses and values.

However Mulchandan discloses

- code labels, data labels referring to data structures, data register names, and index register names; address locations for the code labels and the data labels; and names of the data structures individually expandable to show addresses and values (col.7 lines 41-47 "*Table T-1 is an example of an "Instruction Mode" Bus State Analyzer display. The columns in Table T-1* are "FrU"' *(Frame Number), "A&." (Instruction Address), "Data (normally opcode), "Label", "Codede" and "Bus Cycle". MCUdebug supports a wide range of commands which allow a user* to perform *custom setup of the analyzer"* & col.13 lines 23-30 *"MCUdebug provides a rich set of functions which can allow a user to view analyzer data in variety of ways. The Bus State Analyzer window displayed in Table 1 is an example of the "Instructions Only" view mode of the bus state analyzer data. In the right section of the window (under the label "Bus Cycle") the data displayed shows the actual bus cycle that occurred on the execution of that particular instruction."*& col.12 lines 20-27 & see Table T-1).

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Bade with the teachings of Mulchandan to include code labels, data labels referring to data structures, data register names, and index register names; address locations for the code labels and the data labels; and names of the data structures individually expandable to show addresses and values in order to provide some information close to console from debugger and allow a user to observe the state of the application at any point (col.4 lines 32-40).

8. Per claim 2 (Currently Amended)

the rejection of claim 1 is incorporated and Bade further discloses

- displaying source code associated with a symbol in the fourth screen selected by user ([0101] *"Referring to **FIG. 15,** in a preferred embodiment the graphical object symbols may be selected from a menu **1520,** a textual portion of the object input by the user, and the graphical object connected to other graphical objects using connectors 1505.").*

9. Per claim 3 (Currently Amended)

the rejection of claim 2 is incorporated and Bade further discloses

- displaying a view source button in the fourth screen configured to be activated by a computer mouse to view source code associated with symbol (*As shown in **FIG.15** or toolbar shown in **FIG.25,** the button **C** is for generate C++ code* when user click it by mouse).

10. Per claim 5 (Currently Amended)

the rejection of claim 1 is incorporated and Bade further discloses

- displaying the symbolic information in the fourth screen without typing by the user ([0101] *"Referring to **FIG. 15**, in a preferred embodiment the graphical object symbols may be selected from a menu **1520**, a textual portion of the object input by the user, and the graphical object connected to other graphical objects using connectors 1505."*).

11. Per claim 7 (Original)

the rejection of claim 1 is incorporated and Bade further discloses

- displaying a device enabling expansion of the displayed symbolic information ([0129] *"**FIG. 16** shows a Block construct containing a single Process construct, two Block constructs and a Declaration construct."* & *[0131] "In the example, the interrupt controller waits for an interrupt signal, as sent by one of the two peripheral devices. The Symbol for a Signal-In construct is a rectangle with an arrow pointing inward as either its left or right side."*).

12. Per claim 12 (Currently Amended)

the rejection of claim 1 is incorporated and Bade further discloses

- the programmable circuit includes a processor ([0021] *"The IDE includes a graphical user interface and a design language for forming finite state machine models of hardware components that are coupled to processor simulators, preferably instruction set accurate simulators of processor cores."*).

13. Per claim 13 (Currently Amended)

Bade discloses

An embedded firmware development system, comprising:

- a control module to control the system ([0084] *"**FIG. 4A** is a block diagram illustrating major software modules of a preferred embodiment of the IDE. Further details on the interfaces coupling the software modules are described below in more detail. The software modules of IDE may reside on a memory of a computer having a computer processor and memory, such as personal computer or computer coupled to a network server."*)
- a device interface module coupled to the control module to communicate with a device to be programmed by the system ([0086] *"A simulation loader module invokes other IDE modules when a simulation of an embedded system is initiated from GUI module."*).
- an assembler module coupled to the control module to assemble source code ([0145] *"In one embodiment, the models are hand-written CIC++ models for a particular processor core, with assembly routines for the critical portions to improve speed of performance."*).
- a main module coupled to the control module to display a high-level function screen ([0101] *"As shown in **FIG.**21 the IDE preferably has a menu-driven graphical user interface that preferably includes a design window for creating a design with toolbars for accessing functions using a computer mouse or similar interface. The IDE preferably includes a peripheral design editor and simulator that is adapted to permit hardware IP components and processes to be created and linked with other IP components."*).

- a source module coupled to the control module to display source code for at least two firmware programs ([0085] *"code generator module, and compiler module may be coupled to GUI software module as a hardware design and modeling module for forming a description of the hardware partition of an embedded system."*).
- a hardware resource module coupled to the control module to display hardware resources associated with the device to be programmed ([0085] *"An editor and debugger module, design database module, code generator module, and compiler module may be coupled to GUI software module as a hardware design and modeling module for forming a description of the hardware partition of an embedded system."*).
- a speedbar module coupled to the control module to display symbolic information associated with the source code in one screen ([0085] *"a graphical user interface (GUI) software module for a user to interact with IDE"*).

The symbolic information comprising:

- listings including named registers, data labels for word, byte and short entities ([0163] "*As shown in **FIG. 26, a** Test Bench Builder Toolbar is preferably included to represent the test bench controls, such as a LED, LCD, memory viewer, ASCII terminal window, resource meter, or signal button, and the test bench builder, allows a user to quickly add these controls to a test bench.*" the *memory viewer* may view data words, short words and byte values; and *register viewer* may list information associated with registers).

But Bade does not disclose

- code labels, data labels referring to data structures, data register names, and index register names; address locations for the code labels and the data labels; and names of the data structures individually expandable to show addresses and values.

However Mulchandan discloses

- code labels, data labels referring to data structures, data register names, and index register names; address locations for the code labels and the data labels; and names of the data structures individually expandable to show addresses and values (col.7 lines 41-47 "*Table T-1 is an example of an "Instruction Mode" Bus State Analyzer display. The columns in Table T-1* are "FrU"' *(Frame Number), "A&." (Instruction Address), "Data (normally opcode), "Label", "Codede" and "Bus Cycle". MCUdebug supports a wide range of commands which allow a user* to perform *custom setup of the analyzer"* & col.13 lines 23-30 *"MCUdebug provides a rich set of functions which can allow a user to view analyzer data in variety of ways. The Bus State Analyzer window displayed in Table 1 is an example of the "Instructions Only" view mode of the bus state analyzer data. In the right section of the window (under the label "Bus Cycle") the data displayed shows the actual bus cycle that occurred on the execution of that particular instruction."*& col.12 lines 20-27 & see Table T-1).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Bade with the teachings of Mulchandan to include code labels, data labels referring to data structures, data register names, and index register names; address locations for the code labels and the data labels; and names of the data structures individually expandable to show addresses and values in order to provide

some information close to console from debugger and allow a user to observe the state of the application at any point (col.4 lines 32-40).

14. Per claim 15 (Currently Amended)

the rejection of claim 13 is incorporated and Bade further discloses

- the device includes a processor ([0021] *"The IDE includes a graphical user interface and a design language for forming finite state machine models of hardware components that are coupled to processor simulators, preferably instruction set accurate simulators of processor cores."*).

15. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bade, in view of Mulchandan and further view of van Hoff et al. (U.S. Patent No. 5,778,231 hereinafter "Hoff").

16. Per claim 9 (Currently Amended)

the rejection of claim 1 is incorporated

Both Bade and Mulchandan do not disclose

- parsing the source code to create the listings in the fourth screen.

But Hoff discloses

- parsing the source code to create the listing in the fourth screen (col.2 lines 12-17 *"The inventive compilation method for compiling program source code on a computer to generate compiled code includes identifying symbol references in the source code*

sequentially as the symbolic references occur in the source code, and parsing the code during the compilation to identify each symbol that references another program.").

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Bade and Mulchandan and further include parsing the source code to create the listings in the fourth screen by teachings of Hoff in order to identify externally defined symbols so that the compiler can determine whether the symbols is reference to a remotely located file or to a locally stored file. (Hoff, col.5 lines 42-49).

17. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bade, in view of Mulchandan, Hoff and further view of Hall et al. (US Patent No.4, 720,778 hereinafter "Hall").

18. Per claim 10 (Original)

the rejection of claim 9 is incorporated

Bade, Mulchandan and Hoff do not disclose

- outputting symbolic information for a data structure recursively until resultant fields are no longer structures.

However Hall discloses

- outputting symbolic information for a data structure recursively until resultant fields are no longer structures (col.13 lines 36-38 "*Values of important variables can be seen at each level of a recursive procedure; this is especially useful if a procedure is stuck in infinite recursion.*").

- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Bade, Mulchandan and Hoff and further include outputting symbolic information for a data structure recursively until resultant fields are no longer structures by the teachings of Hall in order to trace the values of data at the entry and exit points of procedure. (Hall, col.13 lines 23-24).

19. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bade, in view of Mulchandan, and further view of Smith et al. (U.S. Patent No. 6,311,324 B1 hereinafter "Smith").

20. Per claim 11 (Original)

the rejection of claim 1 is incorporated

Both Mulchandan and Bade do not disclose

- displaying the symbolic information for particular regions of the source code

But Smith discloses

- displaying the symbolic information for particular regions of the source code (col.4 lines 36-39 "*a tuning program proceeds to analyze application code modules to identify critical regions called hotspots, and displays a graphical view of every hotspot in a module*").
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Bade and Mulchandan and further include displaying the symbolic information for particular regions of the source code by

the teachings of Smith in order to help the user to analyze the region. Once the region has been identified and analyzed, the program advises the user on how to rewrite the program code to improve the performance of the overall application. (Smith, col.3 lines 4-9).

21. Claims 16-18, and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bade, in view of Mulchandani, Hoff and further view of Hall.

22. Per claim 16

Bade discloses

- A method of displaying embedded firmware program information ([0021] "An integrated design environment (IDE) is disclosed for simulating embedded systems")

Comprising:

- displaying a first screen to interact with a user for high level function selections ([0101] *"As shown in **FIG.**21the IDE preferably has a menu-driven graphical user interface that preferably includes a design window for creating a design with toolbars for accessing functions using a computer mouse or similar interface. The IDE preferably includes a peripheral design editor and simulator that is adapted to permit hardware IP components and processes to be created and linked with other IP components."*).
- displaying a second screen to show hardware resources for a programmable circuit ([0099] *"As shown in **FIGS. 28, 29,** and 37, the use of an instruction set accurate simulator to model a processor core permits the processor simulator to exchange memory transactions with the hardware partition and to receive interrupts from the*

hardware partition using APIs linking the hardware partition and the instruction set accurate simulator.").

- displaying a third screen to show source code for a plurality of source code programs to control the programmable circuit ([0173] *"The IDE preferably features a high-quality C++ code generator, hiding all the details of generating simulation code..."*)
- displaying a fourth screen to show symbolic information associated with the displayed source code ([0104] *"**FIG. 36B** shows an example of a software debugger interface window **3685** superimposed over a design window **3620** of a virtual embedded system." & In Fig. 36B shows the design window 3620 associated with source code that is debugging in debugger window 3620*).

The symbolic information comprising:

- listings including named registers, data labels for word, byte and short entities ([0163] "*As shown in **FIG. 26, a** Test Bench Builder Toolbar is preferably included to represent the test bench controls, such as a LED, LCD, memory viewer, ASCII terminal window, resource meter, or signal button, and the test bench builder, allows a user to quickly add these controls to a test bench.*" the *memory viewer* may view data words, short words and byte values; and *register viewer* may list information associated with registers).

But Bade does not disclose

- code labels, data labels referring to data structures, data register names, and index register names; address locations for the code labels and the data labels; and names of the data structures individually expandable to show addresses and values.

However Mulchandan discloses

- code labels, data labels referring to data structures, data register names, and index register names; address locations for the code labels and the data labels; and names of the data structures individually expandable to show addresses and values (col.7 lines 41-47 "*Table T-1 is an example of an "Instruction Mode" Bus State Analyzer display. The columns in Table T-1* are "FrU"' *(Frame Number), "A&." (Instruction Address), "Data (normally opcode), "Label", "Codede" and "Bus Cycle". MCUdebug supports a wide range of commands which allow a user* to perform *custom setup of the analyzer"* & col.13 lines 23-30 *"MCUdebug provides a rich set of functions which can allow a user to view analyzer data in variety of ways. The Bus State Analyzer window displayed in Table 1 is an example of the "Instructions Only" view mode of the bus state analyzer data. In the right section of the window (under the label "Bus Cycle") the data displayed shows the actual bus cycle that occurred on the execution of that particular instruction."*& col.12 lines 20-27 & see Table T-1).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Bade with the teachings of Mulchandan to include code labels, data labels referring to data structures, data register names, and index register names; address locations for the code labels and the data labels; and names of the data structures individually expandable to show addresses and values in order to provide some information close to console from debugger and allow a user to observe the state of the application at any point (col.4 lines 32-40).

Both Bade and Mulchandan do not disclose

- parsing the source code to create the listings in the fourth screen.

But Hoff discloses

- parsing the source code to create the listing in the fourth screen (col.2 lines 12-17 *"The inventive compilation method for compiling program source code on a computer to generate compiled code includes identifying symbol references in the source code sequentially as the symbolic references occur in the source code, and parsing the code during the compilation to identify each symbol that references another program."*).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Bade and Mulchandan and further include parsing the source code to create the listings in the fourth screen by teachings of Hoff in order to identify externally defined symbols so that the compiler can determine whether the symbols is reference to a remotely located file or to a locally stored file. (Hoff, col.5 lines 42-49).

Bade, Mulchandan and Hoff do not disclose

- outputting symbolic information for a data structure recursively until resultant fields are no longer structures.

However Hall discloses

- outputting symbolic information for a data structure recursively until resultant fields are no longer structures (col.13 lines 36-38 *"Values of important variables can be seen at each level of a recursive procedure; this is especially useful if a procedure is stuck in infinite recursion."*).
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Bade, Mulchandan and Hoff and further

include outputting symbolic information for a data structure recursively until resultant fields are no longer structures by the teachings of Hall in order to trace the values of data at the entry and exit points of procedure. (Hall, col.13 lines 23-24).

23. Per claim 17 (Original)

the rejection of claim 16 is incorporated and Bade further discloses

- displaying source code selected by user ([0101] *"Referring to **FIG. 15**, in a preferred embodiment the graphical object symbols may be selected from a menu **1520**, a textual portion of the object input by the user, and the graphical object connected to other graphical objects using connectors 1505.").*

24. Per claim 18 (Currently Amended)

the rejection of claim 16 is incorporated and Bade further discloses

- displaying the source code in the fourth screen selected by the user by clicking on a view source button (*As shown in **FIG.15** or toolbar shown in **FIG.25**, the button **C** is for generate C++ code* when user click it by mouse).

25. Per claim 20 (Currently Amended)

the rejection of claim 16 is incorporated and Bade further discloses

- displaying the symbolic information in the fourth screen without typing by the user ([0101] *"Referring to **FIG. 15**, in a preferred embodiment the graphical object symbols*

*may be selected from a menu **1520,** a textual portion of the object input by the user, and the graphical object connected to other graphical objects using connectors 1505.").*

26. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bade et al., in view of Mulchandani, Hoff, and Hall and further view of Smith.

27. Per claim 24 (Currently Amended)

the rejection of claim 16 is incorporated

Bade, Mulchandan, Hoff and Hall do not disclose

- displaying the symbolic information for particular regions of the source code in the fourth screen

But Smith discloses

- displaying the symbolic information for particular regions of the source code in the fourth screen (col.4 lines 36-39 "*a tuning program proceeds to analyze application code modules to identify critical regions called hotspots, and displays a graphical view of every hotspot in a module*").
- Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine teachings of Bade, Mulchandan, Hoff and Hall and further include displaying the symbolic information for particular regions of the source code in the fourth screen by the teachings of Smith in order to help the user to analyze the region. Once the region has been identified and analyzed, the program advises the user on

how to rewrite the program code to improve the performance of the overall application. (Smith, col.3 lines 4-9).

Response to Arguments

Applicant's arguments with respect to claims 1, 13, 16 and 18 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Junchun Wu whose telephone number is 571-270-1250. The examiner can normally be reached on 8:00-17:00 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.[1]

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JW

WEI ZHEN
SUPERVISORY PATENT EXAMINER

1